HW1-2, due Sept 16 at 9am

In this assignment, you will be ask to perform multiple tasks in C++, Python, and/or matlab. In later assignments, you will be free to use only one language of your choice, and will not be asked to use all three. You may be asked about all three languages on exams.

This assignment requires you submit many small scripts (.cpp, .py, .m). You may upload your solutions to your own github page, and give me access to your github repository for this class. Alternatively, you may upload each problem's solution(s) as a zip file to the Teams page. Do not upload all files to Teams individually, as this will be difficult to manage when grading.

If you are having a hard time with the syntax of some language, it is wise to ask google how to do what you want to do. You will often find an answer quickly.

1. **5 points**. Garcia 1.24 (slightly modified). In this problem, you should use C++, Matlab, and Python, producing source code and 3 figures on log-log axes. For $a = b = 1$, and with $x_0 = 0.1$, the exact solution to the logistic equation is $x(t) = 1/(1 + 9e^{-t})$. Compute $\delta(\Delta t) = |x'(t) - \frac{x(t+\Delta t)-x(t)|}{\Delta t}|$ for $t = 2$ and for $\Delta t = 10^{-n}$.

2. **5 points**. Garcia 1.10 (modified). Machine numbers have finite size, and not all languages have the same max sizes.

   (a) In C++, determine the largest power of 2 that can be represented of `int` type. Do so by defining `int x=1`, and iteratively multiplying by 2 until something goes wrong. Describe what happened to indicate a problem.

   (b) In C++, determine the largest power of `2` that can be represented of `double` type. Do so by defining `double x=1.0`, and iteratively multiplying by `2.0` until something goes wrong. Describe what happened.

   (c) In Matlab, define `x=1` and iteratively multiply by `2` until something goes wrong. Describe what happened.

   (d) In Matlab, define `x=1.0` and iteratively multiply by `2.0` until something goes wrong. Did this fail at the same power of 2 as in (c)?

   (e) In Python, repeat the calculation in (c-d). Since both use weak typing, is the behavior in Python the same as in Matlab?

3. **5 points**. Garcia 1.14 (slightly modified) Gram-Shmidt orthogonalization (GSO) produces two orthogonal vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ from two arbitrary vectors $\mathbf{a}$ and $\mathbf{b}$, by setting $\mathbf{v}_1 = \mathbf{a}$ and $\mathbf{v}_2 = \left[\mathbf{b} - \frac{(\mathbf{a}\cdot\mathbf{b})}{|\mathbf{a}|^2}\mathbf{a}\right] c$, where c ensures $|\mathbf{v}_2| = |\mathbf{b}|$. Implement GSO for two 3-dimensional vectors as a function in C++, Matlab, and Python. You may implement your data structures as arrays or vectors in C++, lists or numpy arrays in Python, and as arrays in matlab.

4. **6 points**. The scripts `c_errors.cpp` and `py_errors.py` (uploaded to Teams) are intended to produce a 10×10 matrix $m_{ij} = |i-j|^2$ and print the elements to the terminal. However, both scripts suffer from common programming errors. Debug both scripts so that they produce the correct output.

5. **7 points**. The dynamics of the approach of the logistic map to a fixed point or attractor may be of interest in many contexts. In this problem, you may use your preferred language for this problem (you don't need to use all 3), and are free to use any library or modify any

source code you wish. If you are using the code from class, `logistic0.x` (where `.x` is your preferred language's extension) might be the most natural place to start.

Plot the first 50 iterations of the logistic map, starting at $x = 0.01$, for $r = 2.00$ and 2.99 on the same figure. The data from each value of r should be connected with a line, be marked by a different symbol, and have a different color. These values of r have a single stable fixed point. Is the fixed point approached reached immediately or does the logistic map oscillate around the fixed point?

6. **7 points**. It is not necessarily obvious if a nonlinear map will exhibit chaos, as you will demonstrate in this problem. You may use your preferred language (you don't need to use all 3), and can use any code from class or from a library you are familiar with. If you use code from class, `logistic1.x` is the most natural place to start.

   (a) Show analytically that the map $x_{n+1} = r\frac{x_n}{1+x_n^2}$ has a stable fixed point for all $r > 1$.

   (b) Compute the bifurcation diagram for the map $x_{n+1} = r\frac{x_n}{1+x_n^2}$ for $1 \leq r \leq 15$ using any numerical method you like. Comment on whether your numerical results are consistent with (a). You should provide source code and a figure of the bifurcation diagram with labeled axes in the solutions.

   (c) Compute the bifurcation diagram for the map $x_{n+1} = r\frac{x_n}{1+x_n^4}$ for $1 \leq r \leq 15$. You *do not* need to analytically compute stability for this map, as it is quite tedious. Comment on any similarities or differences you see for the logistic bifurcation diagram or the results from (b).

7. **5 points**. Garcia 1.14 (slightly modified). Natural units for solving Schrödinger's equation for an electron in a potential are units of eV for energy, the electron mass m_e for the unit mass, and taking $\hbar = 1$. In this case, we would simply solve the differential equation $\left[-\frac{1}{2}\frac{\partial^2}{\partial x^2} + V(x)\right]\Psi = i\frac{\partial}{\partial t}\Psi$. What are the units of distance and time in this simulation? Put another way, what does the dimensionless position $x = 1$ or dimensionless time $t = 1$ correspond to in the physical units of m or s?